Amendments to the Claims

This listing of claims will replace all prior versions, and listings, of claims in the application:

1. (Currently amended) A method of managing memory, comprising:
issuing a data request to remove data;
determining whether the data is being removed from a dirty cache line in a cache memory; and
~~determining whether the data being removed is stack data; and~~
if the data being removed corresponds to a predetermined word in the dirty cache line, queuing the dirty cache line for replacement and not writing the dirty cache line to a memory external to a processor.
~~varying the memory management policies depending on whether the data being removed corresponds to a predetermined word in the cache line.~~

2. (Currently amended) The method of claim 1, wherein the predetermined word is the first word in the dirty cache line.

3. (Currently amended) The method of claim 2, wherein the dirty cache line is invalidated.

4. (Currently amended) The method of claim 2, wherein the dirty cache line ~~holding the stack data~~ is queued for replacement by a replacement policy when a read hit occurs on the first word of the dirty cache line.

5. (Original) The method of claim 4, wherein the replacement policy is a least recently used (LRU) policy.

6. (Currently amended) The method of claim 1, wherein the predetermined word is the last word in the dirty cache line.

7. (Currently amended) The method of claim 1, wherein queuing the dirty cache line for replacement comprises designating the dirty cache line as least-recently-used (LRU) ~~the cache line is a dirty cache line~~.

8. (Currently amended) The method of claim ~~7~~1, further comprising~~,~~ invalidating the dirty cache line if the predetermined word in the dirty cache line is the first word.

9. (Currently amended) A system, comprising:

a memory;

a controller coupled to the memory; and

a stack that exists in the memory;

wherein the memory further comprises a cache memory and a main memory; ~~and~~

wherein, if data being removed comprises a predetermined word in a dirty cache line, the controller queues the dirty cache line to be overwritten, and wherein the dirty cache line is not saved to the main memory.

~~wherein the controller adjusts its management policies depending on whether data that is being removed corresponds to a predetermined word in a cache line.~~

10. (Currently amended) The system of claim 9, wherein the predetermined word is the first word in the dirty cache line.

11. (Currently amended) The system of claim 10, wherein the ~~cache line is invalidated~~controller queues the dirty cache line to be overwritten by invalidating the dirty cache line.

12. (Currently amended) The system of claim 11, wherein the invalidated cache line is queued ~~for replacement by a~~to be overwritten by a least-recently-used (LRU) replacement policy.

13. (Currently amended) The system of claim ~~12~~9, wherein the ~~replacement policy is LRU~~controller queues the dirty cache line to be overwritten by designating the dirty cache line as the least-recently-used (LRU) cache line.

14. (Currently amended) The system of claim 9, wherein the predetermined word is the last word in the dirty cache line.

15. (Canceled).

16. (Currently amended) The system of claim ~~15~~9, wherein the dirty cache line is invalidated if the predetermined word in the dirty cache line is the first word.

17. (Currently amended) A system, comprising:

a processor that executes stack-based instructions;

a cache controller coupled to the processor; and

a cache memory coupled to and controlled by said cache controller, said cache memory storing at least a portion of a stack, said stack having a top and a read access of the stack causes the top of the stack to be read;

wherein, if the processor reads a value from the top of the stack that comprises a word at a predetermined location within a dirty cache line in said cache memory, the cache controller queues said dirty cache line for replacement, and the dirty cache line is not written to a memory external to the processor.

~~wherein the controller invalidates an entire line of said cache memory upon the processor reading a value from the top of the stack if said value from the top of the stack comprises a word at a predetermined location with said line.~~

18. (Previously presented) The system of claim 17 wherein said predetermined location is selected from a group consisting of the first word and the last word of the line.

19. (New) The system of claim 17, wherein the cache controller queues the dirty cache line for replacement by designating the dirty cache line as a least-recently-used (LRU) cache line.

20. (New) The system of claim 17, wherein the cache controller queues the dirty cache line for replacement by invalidating the dirty cache line.